TITLE: Soft Computing in Multidisciplinary Aerospace Design - New Directions for Research

Soft Computing in Multidisciplinary Aerospace Design - New Directions for Research

Prabhat Hajela
Mechanical Engineering, Aeronautical Engineering & Mechanics, Rensselaer Polytechnic Institute
5020 Jonsson Engineering Center, RPI Troy, New York, NJ 12180-3590, United States

Abstract

There has been increased activity in the study of methods for multidisciplinary analysis and design. This field of research has been a busy one over the past decade, driven by advances in computational methods and significant new developments in computer hardware. There is a concern, however, that while new computers will derive their computational speed through parallel processing, current algorithmic procedures that have roots in serial thinking are poor candidates for use on such machines - a paradigm shift is required! Among new advances in computational methods, soft computing techniques have enjoyed a remarkable period of development and growth. Of these, methods of neural computing, evolutionary search, and fuzzy logic have been the most extensively explored in problems of multidisciplinary analysis and design. The paper will summarize important accomplishments to-date, of neurocomputing, fuzzy-logic, and evolutionary search, including immune network modeling, in the field of multidisciplinary aerospace design.

Introduction

The optimal design of multidisciplinary systems has received considerable recent attention [1-3]. This initiative has been largely motivated by a recognition that the design and development of a complex engineering system can no longer be conducted by handling its different components in isolation. Both an increased level of complexity within the components, and the competition driven need to extract the advantages of a synergistic design process, dictate the need for a more comprehensive strategy. Inspite of an increased awareness of the potential of formal optimization methods in the design process, their application in real-life multidisciplinary systems continues to be a challenge. These challenges stem from the fact that practical scale multidisciplinary design optimization (MDO) problems are characterized by the presence of a large number of design variables and constraints, and analysis from various contributing disciplines which are not completely independent, involving one- or two-way couplings between the disciplines. In a number of situations, these analysis may be either very expensive, or in a worse case, simply unavailable to a design engineer in a form other than physical experimentation. MDO can be defined as a multifaceted entity with many conceptual components, and comprising of many established disciplines and technologies. Sobieski [1] defines the principal building blocks of the MDO problem in Figure 1. This taxonomy chart clearly shows the different areas in which research must be focussed if multidisciplinary optimization is to embraced as a standard procedure in industry design practice. These areas can be identified as follows.

- Approximations are needed to replace expensive exact analysis, where the latter can come from either sophisticated numerical codes or through physical experiments. Towards this end, the focus has resided in developing derivative-based extrapolations, or derivative-free approximations such as response surfaces, fuzzy-logic, and neural networks. In case of the latter, design of experiment approaches such as Taguchi methods have an important role in determining the data that must be used to develop such approximations.
- Mathematical modeling is critical to the success of the MDO approach. Such a model is required to predict the system behavior and its sensitivity to design variable changes. Models can be considered to belong to the broad categories of either physical or non-physical models. These models are necessary to answer 'what-if' questions.
- Design-oriented analysis basically deals with the issue of cost vs. accuracy in analysis for optimization. Inexpensive re-analysis is critical and is often based on a linear representation of the problem or on the basis of response-surface like approximations. In some cases, mathematical models can be constructed which only predict the gross system behavior, and provide crude models for preliminary design. Design oriented analysis must also consider how different components of the MDO problem must be executed so as to minimize the effort required in computing their interactions.
- Decomposition is a major concern in MDO, and entails the break-up of a large-scale problem into a sequence of smaller, more tractable subproblems. Depending upon the problem under consideration, the interactions between the decoupled subproblems may be either hierarchic or non-hierarchic.
- Design space search is a major component in MDO and is carried out by using traditional mathematical programming algorithms or newly emergent search techniques based on random sampling such as genetic algorithms and simulated annealing.
- Human intervention continues to be an important issue in MDO as the solution process is not a push-button operation. Operations such as algorithm termination, problem reformulation, interpretation of results, overriding the design process, require human intervention. Methods of artificial intelligence and expert systems have been explored in this context.
- Optimization strategies must continue to be studied in the context of the MDO problem. This includes optimization within a subproblem or a parallel optimization in all subproblems including ways to coordinate the solution towards a converged point.

A detailed discussion of all of these issue is clearly beyond the scope of this paper. This paper attempts to show how soft-computing methods have been adapted in the solution of some of the problems that are endemic to the multidisciplinary design problem. Discussions related to problem decomposition, coordination of solution in decomposed subsystems, design space search, and design oriented analysis, especially in the context of soft computing, are discussed in greater detail.

The MDO Problem

The design of a complex engineering system involves many interacting components or parts. Such interactions are clearly shown in Figure 2 for an aircraft wing structure. The design of this wing structure includes an interaction among the primary disciplines of structures, aerodynamics, controls and propulsion. Also central to this design is the input from the field of aeroacoustics. The figure schematically shows the multiple two-way couplings between the contributing

Paper presented at the RTO AVT Symposium on "Aerodynamic Design and Optimisation of Flight Vehicles in a Concurrent Multi-Disciplinary Environment", held in Ottawa, Canada, 18-21 October 1999, and published in RTO MP-35.

obtained in 8-10 revolutions of the blade. This process of obtaining trim condition prior to load calculation introduces additional computational costs.

The objective of the design problem was to design the blade geometry and internal structure to minimize a weighted sum of hub shear force and bending moments for a hingeless rotor blade in forward flight; aerodynamic, performance and structural design requirements are considered as constraints, and dynamic requirements constitute a multicriterion objective function. The premise behind the approach is that a minimization of the hub loads and moments translates into lower vibrations that are transmitted to the fuselage structure.

The planform geometry of the blade is shown in Fig. 5a. The blade is tapered in both the chord and the depth, and all loads are assumed to be carried by the structural box shown in Fig. 5b. The design variables for this problem are also shown on the figures and summarized in Table 1. The blade planform geometry is defined by the chord ratio λ_c and the point of inception of taper along the span denoted by τ_R.The blade twist varies from q_t at the root to zero at the tip; this variation maybe linear or nonlinear, and is controlled by the twist shape parameter δ. Both q_t and δ were design variables in the problem. Two nonstructural masses were positioned along the span of the blade, and the magnitude of these masses (m_1 and m_2) as well as their locations along the span (d_{m1} and d_{m2}) were considered as design variables. The rotational speed of the rotor Ω was also selected during the design process. The remaining variables, although descriptive of the internal structure of the blade, have a strong influence on both aerodynamic and dynamic performance. These were the horizontal flange thickness t_1, and the left and right vertical sections of the box beam denoted by t_2 and t_3, respectively. For the Graphite/Epoxy rotor blade, the horizontal flanges are symmetric +/-45 deg laminates; this layup is also present on the outer half of both vertical sections of the box beam. The inner half of the vertical walls are divided into two segments, with a layup of +/-θ_1 and +/-θ_2 deg, respectively. This accounts for a total of 14 design variables for the problem. Note that it is relatively easy to increase the problem dimensionality by simply varying the thickness and orientations of plies in discrete segments along the span.

The output quantities of interest are the maximum peak-to-peak values of the scaled shear force, flap bending moment, and lead-lag bending moment (F_z, M_y and M_z), the horsepower required during hover and forward flight (HP_h and HP_f), and the rotor thrust coefficient and solidity (C_T and σ). The last two must be bounded to limit the lift performance of the rotor disk so as to avoid blade stall. Bounds are also required on the blade autorotation (AI) capacity, and on the maximum allowed weight of the blade. A structural failure criterion based on the Tsai-Wu measure was used for the composite structure in this problem. Limiting values of σ_{buck}, the static stresses due to buckling were also imposed. For one given set of design variables, the analysis time required for the evaluation of the objective and constraint functions is substantial (about 18 CPU minutes on a SPARC station) and clearly not amenable to integration within an iterative optimization environment

Both the BP and CP neural networks were used to generate the mapping between the design variables and the response quantities of interest. Some of these mappings are quite nonlinear, and require careful consideration of the choice of network architecture and of the number of training patterns. A number of training patterns was generated, in the range of design variable variation, and this included both stable and unstable designs. A BP network with 14 input layer neurons, 10 hidden layer neurons, and 5 output layer neurons (a 14-10-5 network), corresponding to 5 output quantities, viz., vertical hub shear, flap moment, lag moment, rotor thrust, and the failure criterion index $\overline{R}$, was established and trained with 550 training patterns. This training presented problems in that it was difficult to reduce the training error to below 3%. The training data was then sorted to separate the designs that yielded stable and unstable responses; it was considered expedient at this stage to establish 5 networks, each mapping the design variables into one output only (14-10-1 networks for all but the lag moment, where a 14-10-8-1 network was used). The training of each of these networks could be done in parallel. The training using these separated patterns procceded well, converging to an error of less than 1%, with the exception of the lag moment (error was 2.6%). The observed pattern of time variation for the lag moment was quite nonlinear, and provides an explanation for the discrepancy in training. Table 2 shows the results of testing these networks for generalization performance using 7 sets of design variables that were not part of the original training set. Similar generalization performance was obtained from networks trained with the unstable data. The difficulty in training the networks for the combination of stable and unstable data can be attributed to either a) a completely different input-output relationship in one or more components or b) excessive data for the number of weights and bias constants in the network that could be varied to fit the data.

Similar experiments were also performed using the full CP network, that generates an identity mapping of the type [X,Y] -> [X',Y']. This network architecture requires a much larger number of input patterns, and the quality of generalization depends upon the number of Kohonen layer neurons that are permitted (indirectly a measure of maximum cluster radius). The results of numerical experiments designed to test this network are shown in Table 3.

Fuzzy Logic Based Function Approximations

In contrast to neural networks or other conventional function approximation techniques, fuzzy logic is based on natural languages. In a system, the problem description may be imprecise, not due to randomness, but because of inherent fuzziness. By taking advantage of the significance of natural language which has developed over many years, fuzzy logic can effectively model a complex real world without getting into the unnecessary detail.

The notion of fuzzy sets was first introduced by Zadeh in 1965 [12]. Since then, conceptual ideas were developed for nearly 10 years with very few applications. However, many recent successful uses of fuzzy set theory in various fields have established it as an effective tool to represent and manage vague information [13-15]. Fuzzy set theory has also gained much attention in the field of design optimization. Fuzzy optimal design of structures was introduced by Yuan and Quan [16,17]. Fuzzy optimization techniques were also applied in structural optimization problems with multiple sitivity over the range of network training, and identifies the importance of any input component on an output quantity of interest. Such an analysis can be used to systematically partition the design space in a decomposition based design approach. This concept is discussed in a later section of this paper.

The counterpropagation (CP) neural network was first introduced by Hecht-Nielsen as a combination of two basic architectures - the Kohonen's self-organizing neural network and Grossberg's outstars neurons. However, the original version of the network did not receive widespread attention due to its unimpressive generalization performance, particularly in comparison to the multilayer perceptron model. As shown in Fig. 4, this network contains three layers - a fan-out layer as in the BP network, a layer of Kohonen or feature sensitive neurons, and an interpolating or Grossberg layer. The inputs to the network are directed to the Kohonen layer, which acts like a clustering device. In other words, neurons in this layer classify all input vectors based on some identifiable features in these vectors. Each neuron in the Kohonen layer represents one such cluster, and the interconnection weights between the input nodes and this neuron are representative of an average of all input patterns of that cluster. Similarly, the interconnection weights between each Kohonen neuron and the output or Grossberg layer neurons are representative of an averaged output of all patterns belonging to the cluster. Improvements to this basic format have been introduced wherein an input vector is classified as belonging to many different clusters, albeit to different degrees. A weighted average using the different clusters produces a much better generalization performance.

The primary differences between the CP and the BP model are in the time and data required for network training, and in the performance of their generalization capabilities. The former requires less computational effort to train, an issue of some importance when one considers modeling of extremely large multidisciplinary systems. However, its generalization performance is poorer when compared to the BP model. Improvements to the CP network have been implemented that partially circumvent this problem; however, large sets of training data continue to be required for their effective use. An advantage over the BP network is that CP networks provide a pattern completion capability, wherein, upon presentation of an input vector to the network, some components of which may be missing, the network approximates the missing components to produce a relevant output.

In using neural networks for approximate analysis, a set of input-output training patterns must be obtained from the real process that one is attempting to simulate. Determination of the number of such training pairs, and how they should span the intended domain of training, requires experimentation and experience. The same statement is applicable to the selection of the network architecture, i.e., the number of layers and the number of neurons in such layers. While mathematics can be used to assess bounds on such requirements, in a number of engineering problems, such approximations tend to be over conservative. This continues to be an active area for research.

Neural networks have been explored as function approximation tools in problems of multidisciplinary analysis and design, most commonly as a computationally inexpensive replacement for procedural analysis. In such cases, training data is generated from a procedural simulation of the multidisciplinary system, and a neural network is trained to mimic the input-output relations of this system (generalization). In such use, the neural network may be considered as a response surface approach where the order of the polynomial fitting function does not have to be specified. In fact, the neural network is a special form of response surface where the response function is a nested squashing function; the interconnection weights of the network that have to be learned correspond to the regression parameters in a typical regression model. Such a neural network is ideally well suited for an immersive design model, including one using a virtual reality environment.

ANN in Function Approximations - Example

At a preliminary design level, the multidisciplinary sizing of a rotor blade may require an integration of the disciplines of acoustics, aerodynamics, dynamics, and structural analysis within the optimization framework. This simplified example illustrates some of the inherent complexities in such design problems, and the use of neural networks in this context. The use of composites in rotorcraft blade design provide opportunities for enhanced aerodynamic, structural, and dynamic performance. With composites, it is practical to fabricate non-rectangular blades with variations in twist distribution and airfoil sections along the blade span, thereby contributing to increased flexibility in aerodynamic design. Satisfactory aerodynamic design requires that the required horsepower for all flight conditions not exceed the available horsepower, that the rotor disk must retain lift performance to avoid blade stall, and that the vehicle remain in trim. Important factors in structural design include material strength considerations for both static and dynamic load conditions. A combination of flapwise, inplane, torsion, and centrifugal forces typically comprise the static loading. Another important consideration that encompasses both structural and aerodynamic design, is the autorotation capability. The autorotation requirement pertains to maintaining the mass moment of inertia of the rotor in the rotational plane at an acceptable level. This is a function of the vehicle gross weight, rotor aerodynamic performance, and the rotor system mass moment of inertia. Finally, dynamic design considerations of the rotor blade pertain to the vibratory response of the blade under the applied loads; this design limits the dynamic excitation of the fuselage by reducing the forces and moments transmitted to the fuselage.

A finite element in time and space formulation was used to model the dynamics of the blade [11]. This formulation is based on a multibody representation of flexible structures undergoing large displacements and finite rotations, and requires that the equations of motion be explicitly integrated in time. An unsteady aerodynamic model is used to obtain the induced flow and to calculate the aerodynamic forces and moments in hover and forward flight. In addition to the geometric nonlinearities that are inherent in this problem, the loading on the blade varies as it moves around the azimuth - on the advancing side the flow velocity over the blade is additive to its tangential speed; on the retreating side, these velocities substract. Consequently, in order to maintain force and moment equilibrium, the pitch of the blade is continuously changed as it rotates around the azimuth, and the hub loads are a function of the blade rotational frequency. There is a transient period during which equilibrium of the vehicle is established (trim), and this is generally subsystems. The interactions between the disciplines generally result from interactions between specific physical phenomena, two of which are illustrated in this figure. In such an environment where everything appears to effect everything else, design of subsystems in isolation is clearly not the strategy of choice. In an approach where the coupled multidisciplinary problem is treated as a single, large-scale optimization problem, the following difficulties have been identified:

- The dimensionality of the design space may increase to a degree that obtaining reliable solutions to the optimization problem is rendered questionable. Furthermore, there is a diminished capacity to evaluate the acceptability of solutions in higher-dimensionality design spaces.
- For an iterative analysis environment that is typical of optimization, the presence of coupling between disciplines would introduce an inner loop of iteration (analysis iteration) that adds to the computational costs.
- In a number of design problems, the design space may be nonconvex, and in some situations, even disjointed. Such characteristics call for the use of nontraditional search techniques that do not have a propensity to seek the nearest relative optimum from the nominal starting solution

Decomposition methods are introduced in multidisciplinary optimization to reduce large coupled optimization problems into a sequence of coordinated, smaller, more tractable subsystems. The subsystems not only allow for a reduction in problem dimensionality, but also allow for implementing specialized methods of analysis in each subsystem, and possibility of distributed, parallel processing. The interpretation of optimization results in each subsystem is also facilitated by the dimensionality reduction.

We start first with a generic mathematical statement for the optimization problem written as follows.

$$\begin{aligned} & Minimize && F(X) && \\ Subject\ to\quad & g_jX \le 0 && j = 1, m && \\ & h_k(X)) = 0 && k = 1, p && (1) \\ & X^L \le X \le X^U && && \end{aligned}$$

Here X is the vector of design variables; superscripts 'L' and 'U' denote the lower and upper bounds, respectively; F(X) is the objective function and $g_j(X)$and $h_k(X)$ are the inequality and equality constraints, respectively. If the dimensionality of the aforementioned design problem is manageable (of the order of a few hundred design variables), and if the gradient information is readily available, then traditional gradient based methods of nonlinear programming can be effectively used to obtain the optimal solution. However, in those problems where the design variables are a mix of continuous, discrete, and integer type, gradient information is not very useful, and alternative strategies must be investigated. A mixed-variable design space also limits the usefulness of traditional gradient-based optimization algorithms, and non-gradient methods for optimal search have received attention in this context.

Another dominant concern in large-scale MDO problems is the high computational cost of analysis. Early implementations of multidisciplinary synthesis methods looked towards approximation methods for relief in this area (efficient structural reanalysis and Taylor approximations) [4,5]. While these ideas are still relevant and are broadly described as 'analysis for design', new requirements for developing non-gradient based global approximations have emerged. This latter requirement is in part due to the increasing interest in applying optimization methods to real systems with mixed-variable design spaces. More importantly, it is driven by the need to generate 'mathematical models' for including disciplines such as manufacturability, cost, and maintainability, into the optimization problem. The use of response-surface like approximations based on neural networks and fuzzy logic have been explored in this context.

Approximate Models for Analysis

This section of the paper describes the use of artificial neural networks and fuzzy logic in developing approximate models for analysis, designed primarily to reduce the computational effort involved in MDO problems. The use of fuzzy logic in this application is motivated by the need to include in the optimization process, disciplines for which precise and well-defined analytical models are unavailable; these include, for example, issues pertaining to the cost, manufacture, and maintainence of designed artifacts. Both artificial neural networks and fuzzy logic based models are much like response surfaces, where a polynomial is fitted to given experimental or numeric data. Unlike the response surface approach where the order of the fitted polynomial must be specified, these methods provide greater flexibility to the user.

Among the most widely adapted neural network architectures in function approximation are the backpropagation (BP) network, counterpropagation (CP) network, and the radial basis network [6,7]. As shown in Figure 3, the BP network architecture consists of a layer of artificial neurons to which the external stimuli are presented, a series of hidden layers of artificial neurons, and a layer of neurons at which the output is available. The input neurons do not process the input stimulus; they simply serve as 'fan-out' points for connections to neurons in successive layers. Neurons in each layer are connected to all neurons in adjacent layers; there is an interconnection weight associated with this connection which defines the strength of the connection. Also associated with each artificial neuron is what is referred to as an activation function (sigmoid function or step function). The weighted sum of all inputs to a particular neuron are processed through this nonlinear activation function to produce a neuron output, which then feeds into all neurons of the next layer.

The presence of the hidden layer, and the nonlinear activation functions, enhance the ability of the networks to learn nonlinear relationships between the presented input and output quantities. This 'learning' or 'training' in these networks simply requires the determination of all interconnection weights of the network and characteristics of all activation functions, so that the network accurately produces the desired output for each of the input patterns used in the training. Once such a trained network is established, it responds to a new input within the domain of its training by producing an estimate of the output response. To this extent, it serves as a function approximation tool that provides inexpensive function information in stochastic sampling based search procedures [8]. The trained weights of this network can also be used to identify dependencies among design variables and design objectives/constraints [9,10]. This weight analysis may be considered as a smeared global senobjective functions [18,19].

Fuzzy sets are inherently different from classical sets; while the latter either wholly includes or excludes any given element, a fuzzy set can contain an element partially by using degrees of membership. An example which defines fuzzy sets is as follows. Consider the elements of a set defined as **heights** to include 5'0", 5'8", and 6'2". If another set is to include only the **tall heights** from the given elements, it would, most definitely, include the element 6'2" and exclude the element 5'0" since 6'2" is considered to be tall by most while 5'0" is not. For the element 5'8", it would be difficult to determine whether to include or exclude it from this set. Some will consider a height of 5'8" to be tall and others won't; this element sits on the fence. While classical sets have no way of accommodating this type of element, fuzzy sets can include them by assigning each of them a degree of membership. The degree of membership of each elements in a fuzzy set can be determined by its membership function. Since 5'0" is excluded from the set referred to as "tall", it is assigned a degree of membership of 0.0; in a similar vein, a degree of membership of 1.0 is assigned to the element 6'2", while some numerical value between 0 and 1 is given to the element 5'8". As an example, in fuzzy logic, 5'8" can be assigned a degree of membership of 0.7 which indicates that the person is somewhat tall.

If a classical set **tall heights** of a real number greater than 6 can be expressed as $tall = \{x|x > 6\}$, then a fuzzy set **tall heights** in X is defined as a set of ordered pairs $tall = \{(x, \mu_{tall}(x))|x \in X\}$, where X is the universe of disclosure, x is an element of X, and $\mu_{tall}(x)$ is called the membership function of x in the fuzzy set **tall heights**. A membership function maps each element of x to a value between 0 and 1. Although the membership function can be any arbitrary curve whose shape is defined according to one's subjective perception, several most commonly used parameterized functions are available as a guideline. These include the triangular membership function, the trapezoidal membership function, the Gaussian membership function, the generalized bell membership function, and the sigmoidal membership function. Associated with each are a number of parameters that must be specified; they also offer a variety of smooth or nonsmooth functions to model variations of a quantity of interest.

Fuzzy logic is much like standard boolean logic except for the fact that in addition to 0 and 1, fuzzy logic can also operate with any numerical value between 0 and 1. There are a number of common operations such as intersection, union, and complement of fuzzy sets. The standard truth tables of Boolean logic are extended to fuzzy sets by replacing single-valued operations by multivalued logical operations [12]. Fuzzy rules are defined to map linguistic input and output values, and use conditional statements in the form of if-then-rules. In the case of boolean logic, if the antecedent part of if-then rule is true then the consequent part of if-then rule is also true. However, in fuzzy if-then rules, if the antecedent is partially true to some degree then the consequent is also partially true to that same degree. Given the freedom to choose from among different membership functions, fuzzy logic allows for the creation of an optimally tuned input-output function mapping. Evolutionary fuzzy modeling employs genetic algorithm based optimization to evolve fuzzy rules and membership function parameters. Genetic algorithms select from among discrete choices of membership functions and tune their parameters until the error between fuzzy outputs and target values are minimized.

Fuzzy logic approximation models are most appropriate for those situations when there is imprecise definition of parameters describing the desired model. One such model that has been constructed consists for creating a numerical model that relates the layup sequence of the individual plies in composite panels for rotorcraft fuselage, to the time required to fabrucate such models on an automated fiber lay-up machine. In addition to limited numerical data that relates parameters such as panel geometry, location and frequency of cutouts, and fiber orientations in individual plies to the layup time, input from the machine operator was incorporated in construction of a fuzzy-logic based approximate model that was subsequently used in design optimization. Additional details on this model may be found in Reference [20].

Evolutionary Search and Genetic Algorithms

The strengths of evolutionary algorithms have been clearly established with reference to optimal search in generically difficult but very realistic multidisciplinary design problems such as those containing discontinuities or nonconvexities in function behavior, discrete variation in design variables, and where gradient information is generally unavailable. The genetic algorithm is based on an elitist reproduction strategy, where chromosomal representation of designs is evolved using random operations encompassed in operation like crossover and mutation, with bias assigned to those members of the population that are deemed most fit at any stage of the evolution process. In order to represent designs as chromosome-like strings, stringlike representations of the design variables are stacked head-to-tail. Different representation schemes have been adopted, including use of the decimal values of the design variable [21], use of integer numbers [21], or most popularly, a conversion of all design variables into their binary equivalent. In the latter case where the chromosomal string is a number of 0's and 1's, the numerical precision with which the design variable is represented is determined by the string length.

Increased adaptation into the multidisciplinary design environment has been accompanied by a number of modifications to the basic GA approach. Of these, direct schemes (non-penalty function methods) by which to account for design constraints [22,23], have received some attention. An approach applicable to a case where constraints are linear and the design space convex, has been described in [22]. Other methods, based on strategies that adapt useful features of the feasible designs into the infeasible population, have been proposed [23,24]. In [24], the process of adaptation is through the use of an expression operator, which like the crossover and mutation operations in genetic search, is probabilistic in nature. A similar process of adaptation ("gene-correction" therapy) is also at work in another strategy that is based on immune network simulation [23].

Binary coded GA's search for an optimal design from among a discrete set of design alternatives, the number of which depend upon the length of the chromosomal string. Large number of design variables, and/or considering a very fine precision in continuous design variable representation contributes to long chromosome string lengths which detracts from the efficiency of the search process. These problems are particularly relevant in large-scale MDO problems, and several solution strategies have been proposed in this regard. Methods such as multistage search [25], wherein the granularity of the genetic algorithm based search is varied through a successive increase in the precision, and an approach which assigns significance to the previous generations of evolution in genetic search referred to as directed crossover [25], have been proposed. The latter simply attempts to determine through computations, significant bit positions on the string, and to constrain the crossover operation to these bit locations. A number of applications of both the basic GA and its enhanced forms, in problems of multidisciplinary structural design, structural layout determination, and composites design, are described in [26].

The implementation of genetic algorithms in a decomposition based approach has also been studied [27]. Consider the design problem to be formulated in terms of a design variable vector $\boldsymbol{X}$. Also, let the design constraints $\boldsymbol{g}_j(\boldsymbol{X})$ belong to the global constraint set $\boldsymbol{G}$. The vector $\boldsymbol{X}$ and constraint set $\boldsymbol{G}$ are said to define a system level problem that is formulated as follows:

$$\begin{gathered} Min\ \ or\ \ Max\quad F(X) \\ subject\ \ to\ \ G \equiv \{g_j(X),\ j = 1 \dots NCON\} \leq 0 \end{gathered} \tag{2}$$

Assume further that the best topology for decomposing the problem domain resulted in three subproblems A, B, and C, and the design variables and constraints for each of these subproblems are denoted by X_A, X_B, X_C, and g_A, g_B, and g_C, respectively. The objective function F($\boldsymbol{X}$) for each of the subproblems is the same, and is the system level objective function. The system level problem of eqn. (2) is now represented by the following three subproblems.

$$\begin{gathered} Min\ \ or\ \ Max\quad F(X_A), \\ subject\ \ to\ \ g_A(X_A)\ \ \pounds\, 0,\ X_B,\ X_C = const \\ Min\ \ or\ \ Max\quad F(X_B), \\ subject\ \ to\ \ g_B(X_B)\ \ \pounds\, 0,\ X_A,\ X_C = const \\ Min\ \ or\ \ Max\quad F(X_C), \\ subject\ \ to\ \ g_C(X_C)\ \ \pounds\, 0,\ X_A,\ X_B = const \end{gathered} \tag{3}$$

The principal challenge in this approach is to determine an appropriate topology for problem decomposition, and once such a topology has been established, to develop a procedure for coordinating the solution among the decomposed subproblems. The latter implies that the objective function obtained from each of the three subproblem optimizations be the same when the process converges to an optimal design.

A reasonable and logical approach for partitioning is one where balanced subsets of design variables would be assigned to different subproblems, and where each subproblem would be responsible for meeting the system level design objectives and for satisfying constraints most critically affected by the design variables of that subproblem. A trained BP network, can be used to extract the required causality. The weights of a BP network trained to relate the input design variables to the design constraints can be used to identify dependencies among design variables and design objectives/constraints. This weight analysis may be considered as a smeared global sensitivity over the range of network training, and identifies the importance of any input component on an output quantity of interest. The approach results in the formation of a transition matrix [T], the components T_{ij} of which reflect the importance of the i-th input quantity on the j-th output component. First, the matrix product of the interconnection weight matrices is performed as indicated in Eqn. (4), and the elements of the transition matrix normalized as shown in Eqn. (5).

$$[T] = \prod_{k=1}^{N-1} W^k \tag{4}$$

$$\overline{T_{ij}} = \frac{T_{ij}}{max_j\ |T_{ij}|} \tag{5}$$

In the above, W^k is the k-th weight matrix, the coefficients w_{ij}^{kl} of which represents the interconnection weight between the i-th neuron of the k-th layer and the j-th neuron of the l-th layer; N denotes the total number of layers of neurons in the network architecture. This normalized matrix $\overline{T}_{ij}$ incorporates the effect of the sign of interconnection weights in the analysis. A systematic approach of using this transition matrix to decompose the problem is presented in Reference 27. For a larger version of the rotorcraft blade design problem defined earlier where the number of design variables has been increased to 42, the use of this transition matrix results in a decomposition of this problem as shown in Table 4. The form of partitioning allows for the most effective variables for a particular set of constraints to work in each subproblem optimization. It is also easy to recognize that such a split-up in variables could not be realized by a partitioning based on disciplinary concerns only, such as one that is largely used in current practice.

Once the design problem has been partitioned into a number of subproblems, the solution within each subproblem must proceed with adequate consideration of how a local design change influences the results of analysis in another subproblem. This is referred to as solution coordination and is required due to the fact that the subproblems are seldom completely decoupled. Two strategies that allow for consideration of these couplings are a) an approach based on the use of the counterpropagation (CP) neural network, and b) through the use of the simulation of a biological immune system modeling. As stated earlier, an important property of the CP network is a pattern completion capability - if an incomplete input pattern is presented to the network, the network estimates the most likely make-up of the missing components. This pattern completion capability can be of use in GA based decomposition design, by linking the GA optimizer in each subproblem with a trained CP network. In this mode of operation, the inputs to the CP network in each subproblem are the design variables for that subproblem, and approximations of the best combinations of variables in other subproblems. Details of how the progress of design variables of each subproblem are transmitted to other subproblems are described in [28]. The use of the biological immune system as an approach to communicate the coordination information is discussed in the following section.

Immune Network Modeling

In biological immune systems, foreign cells and molecules, denoted as antigens, are recognized and eliminated by type-specific antibodies. The task of recognizing antigens is formidable due to the very large number of possible antigens; it is estimated that the immune system has been able to recognize at least 10^{16} antigens. This pattern recognition capability is impressive, given that the genome contains only about 10^5 genes. This process can be simulated using the genetic algorithm approach, and has been the subject of considerable study [29,30].

A matching function that measures the similarities between antibodies and antigens, substitutes for the fitness function of the genetic algorithm. In a typical simulation of the immune system, the fitness of an individual antibody would be determined by its ability to recognize specific antigens, and a genetic algorithm using the matching function as a measure of fitness would evolve the gene structure of the antibody in an appropriate manner. In the context of a binary-coded genetic algorithm, the antibodies and antigens can also be coded as binary strings. The degree of match or complimentarity between an antibody and an antigen string indicates the goodness of that antibody. A simple numerical measure $Z = \sum t_i \; i=1,N_{string}$ can be defined, where N_{string} is the length of the binary string, and $t_i = 1$ if there is a match at the i-th location of the two strings, and is 0 otherwise. A larger value of Z indicates a higher degree of match between the two strings. Using a traditional GA simulation, a population of antibodies can be evolved to cover a specified pool of antigens, with Z used as the fitness measure for this simulation. The manner in which this pattern recognition scheme is invoked will determine whether the evolved antibodies are 'specialists', i.e. adapted to specific antigens, or generalists that provide the best match to a number of different antigens. From an applications standpoint, generation of both specialist and generalist antibodies is useful, and some of the applications have been discussed in [31].

In using this approach to account for interactions among temporarily decoupled subproblems, the motivation is to adapt changes in design variables from other subproblems into a particular subproblem with a prescribed frequency in the artificial evolution process. Note that updating the design variables of other subproblems must not simply involve introducing the best design from those subproblems but rather an average of the best few designs. In this regard, a generalist antibody would be developed that is the best representation of a number of good designs.

The proposed decomposition-based design procedure using immune network system may be summarized (see Figure 6) as follows. The stringlike chromosome structure representing the design contains a definition of all design variables. After partitioning of the design variable vector for each subsystem has been performed, genetic evolution is carried out in each subsystem in parallel, with the fitness function described in terms of the system level objective function. It should be noted that only that subsection of the chromosome string which corresponds to the design variables for that particular subsystem is changed. This process can be carried out for a fixed number of generations, and then a predetermined fraction of fit strings from each subsystem are introduced as antigens to all other subsystems. A second stage of genetic search is then performed in each subsystem where the fitness function is defined in terms of each individual's ability to manufacture antibodies that match the newly introduced antigens. This stage of the genetic search is seen as a correction step that introduces compatibility between the different subpopulations, so that the process eventually converges on the desired design.

Computational Intelligence

Evolutionary search algorithms have a significant role in implementing ideas of computational intelligence in multidisciplinary design. Recent research has shown [32] how binary-coded rules in the IF-THEN form can be evolved using the genetic algorithm, based on information derived from a computational domain. In such classifier system type of machine learning approaches the rules may be completely arbitrary in the beginning but evolve into meaningful statements with information extracted from the computational domain. This approach has powerful implications in overcoming problems of a brittle rule-base that were endemic in traditional rule-based systems.

A classifier system is generally divided into two parts, a set of rules or classifiers, and a message list. The message list contains at least one input from the external environment and also provides the framework for the rules to interact (any rules generated internally are posted here as well). Hence, the message list is dynamic in nature, constantly evolving as the system changes. The classifier rules are made up of three distinct segments - conditions, actions and strength. To facilitate the use of genetic algorithms in this approach, all rules are coded as binary strings. The conditions allow the classifier to read the message list by searching for matches between the condition and the message list. If a match is found, the action is posted to the message list. The strength is a number associated with each rule designed to indicate its value to the systems, and forms the basis for learning. If a rule helps bring about useful responses, it gains strength. Similarly, an ineffective rule is weakened and perhaps ultimately purged from the system. The strength of the proposed approach would be to introduce new rules into the system based on principles of genetic search.

Consider for example, a heuristics based optimization procedure where the objective is to minimize the weight of a truss structure while ensuring that the bounds on maximum permissible stress in each element of the truss are not violated. A solution strategy would entail constructing a number of random classifiers in the IF-THEN (Condition-Action) format. The condition segment of the classifier can be chosen in a form that it allows for the classifier to be related to the current state of the design. One approach for doing this would be to construct a composite measure of all constraints of the problem. The action part of the classifier could be the changes required in each component of the design variable. A very simple-minded operation of the classifier would be to perform a match between the classifiers and the current state of the design; this match would be based on both the magnitude and sign of the constraint values. The winning classifier would be allowed to execute its action segment, and the resulting change in the state of the design as indicated by the weight and the constraint values, used to either increase or attenuate the strength of that classifier. Since both the condition and action segment of the classifiers are initially randomized, the first set of design changes may be quite erratic, and the rules must evolve in a manner that subsequent steps will produce improvements in the results. This objective is attained by using a genetic algorithm to evolve the classifiers based on their fitness value as indicated by their current strengths.

The approach has also been used in enhancing the process through which neural networks are used to create function approximations. A rule learning procedure was implemented wherein computational feedback on the performance of a partly trained network was used to determine the amount and location of new training patterns required to improve the network generalization performance [33]. A similar approach to improve the quality of response surface based approximations is presented in [34]. The application of a classifier system approach in turbine design optimization is presented in Reference 35.

New Research Directions

This review would be incomplete without some reference to future directions for exploration. The role of soft computing tools in structural analysis and design is today on firmer ground than it has ever been before. Computing speeds today are in the MFLOPS-GFLOPS range and predictions indicate TFLOPS performance and better in the next decade. Analysis and design techniques must be revisited from a completely new perspective if such hardware is to be used in the most effective manner. Current algorithms, through all manners of software enhancement and efforts to parallelize, have their origins in serial thinking, and without the required intrinsic parallelism, are victims of the law of diminishing returns when placed on parallel machines. A completely new line of thinking born in the parallel processing environment is required for developing the next generation of structural analysis and design tools, and here is where soft computing tools are likely to have the most visible impact. What must result from new developments are real-time methods to analyze a design artifact - these tools should provide a design engineer with an immersive design capability in which not only search directions and step sizes are computed rapidly, but what-if questions pertinent to the design are available instantaneously. Research targeting these lofty goals has begun, albeit at very rudimentary levels. Specific goals of such research which spans the spectrum of soft computing tools includes the following

Use of neural networks and fuzzy-logic tools to develop response surface-like approximations for discrete and discontinuous functions which are significant in the context of topological design of structural systems, or in an immersive/ interactive analysis and design mode, where exploration of discrete concepts is most critical. The concept of using macro-neural networks (neuroelements) which can be adaptively interconnected during the evolution of design, has been considered. Efforts to expand the use of evolutionary algorithms to large-scale design problems through more efficient implementations in a parallel computing environment are worthy of consideration Similarly, extending these algorithms to better handle fuzzy design constraints and objective criteria through an effective integration with fuzzy-logic techniques must be explored. These are particularly relevant in an environment where manufacturing considerations must be included in the design cycle. Finally, an exploration of entirely different computational paradigms such as cellular automata [36] to both analyze and design structural systems presents distinct possibilities. This approach represents a departure from the traditional procedural analysis. Instead of using fundamental equations of physics pertinent to a problem to analyze a domain, the idea here is to decompose the problem domain into a number of grids, where the property of each cell within this grid evolves or emerges through an interaction with the surrounding cells in the grid. This self-emergent or self-organizing behavior is thought to be significant in the development of the next generation of structural synthesis tools; it is an intrinsically decentralized computational paradigm ideal for multiple parallel processor machines. Preliminary results of work in this area are reported in [37,38].

Closing Remarks

The paper has described a subset of attempted applications of soft-computing tools in problems of multidisciplinary analysis and design. These tools represent significantly improved capabilities for solving generically difficult problems; more specifically, they overcome difficulties related to problem dimensionality, handling of a mix of discrete, integer, and continuous design variables, accounting for discontinuities or nonconvexities in the design space, and improved capabilities for modeling and design in the absence of gradient information. The last item is particularly relevant in a design for manufacturing environment, where manufacturing and production related constraints have received increased attention. Soft computing methods are expected to have a major role in the development of the next generation of tools for multidisciplinary analysis and design.

References

1. Sobieszczanski-Sobieski, J., "Multidisciplinary Design Optimization: An Emerging New Engineering Discipline", World Congress on Optimal Design of Structural Systems, Rio de Janeiro, Brazil, August 2-6, 1993.

2. Tolson, R.H. and Sobieszczanski-Sobieski, J., "Multidisciplinary Analysis and Synthesis: Needs and Opportunities", AIAA Paper No. 85-0584, 1985.

3. Abdi, F., Ide, H., Levine, M., and Austel, L., "The Art of Spacecraft Design: A Multidisciplinary Challenge", 2nd NASA/Air Force Symposium on Recent Advances in Multidisciplinary Analysis and Optimization, NASA CP-3031, Sep. 1988.

4. Noor, A.K., and Lowder, H.E., 1974. "Approximate Techniques of Structural Reanalysis", Computers and Structures, Vol. 4, pp. 801-812.

5. Schmit, L.A., and Farshi, B., 1974. "Some Approximation Concepts for Structural Synthesis", AIAA Journal, 12, 5, 692-699.

6. Hajela, P., Neural Networks - Applications in Modeling and Design of Structural Systems, CISM Lecture Notes, Udine, Italy, October 1998.

7. Haykin, S., 1994. Neural Networks - A Comprehensive Foundation, Macmillan Publishing Company, Englewood, New Jersey.

8. Hajela, P., and Lee, J., "Role of Emergent Computing Techniques in Multidisciplinary Design", proceedings of the NATO Advanced Research Workshop on Emergent Computing Methods in Engineering Design, Nafplio, Greece, August 1994, Springer Verlag, 1996, pp. 162-187, eds. D. Grierson and P. Hajela.

9. Goel, S. and Hajela, P., "Classifier Systems in Design Optimization", proceedings of the AIAA/ASME/ASCE/ AHS SDM meeting, Kissimee, Florida, April 1997.

10. Hajela, P and Szewczyk, Z., "Neurocomputing Strategies in Structural Design - On Analyzing Weights of Feedforward Neural Networks", Structural Optimization, Vol. 8, No. 4, pp. 236-241, December 1994.

11. Bauchau, O. A., and Kang, N.K..: "A Multibody Formulation for Helicopter Structural Dynamic Analysis", Journal of American Helicopter Society, Vol. 38, No. 2, pp. 3-14, April 1993.

12. Zadeh, L., "Fuzzy Sets," *Information and Control*, Vol. 8, pp. 338-353, 1965.

13. Uragami, M., Mizumoto, M., and Tanaka, K., "Fuzzy Robot Controls," Journal of Cybernetics, Vol. 6, pp. 39-64, 1976.

14. Dubois, D., "An Application of Fuzzy Set Theory to Bus Transportation Network Modification," Proceeding of the Joint Automatic Controls Conference, Philadelphia, 1978.

15. Kacprzyk, J., "Fuzzy Set-Theoretic Approach to the Optimal Assignment of Work Places," in G. Guardabassi and A. Locatelli (eds), Large Scale Systems Theory and Applications, Udine, pp. 123-131, 1976.

16. Yuan, W. G. and Quan, W. W., "Fuzzy Optimum Design of Structures," Engineering Optimization, Vol. 8, pp. 291-300, 1985.

17. Yuan, W. G. and Quan, W. W., "Fuzzy Optimum Design of Aseismic Structures," Earthquake Eng. Sturct. Dyn., Vol. 13, pp. 827-837, 1985.

18. Rao, S. S., "Multi-Objective Optimization of Fuzzy Structural Systems," International Journal for Numerical Methods in Engineering, Vol. 24, pp. 1157-1171, 1987.

19. Yoo, J. and Hajela, P., "Multicriterion Design of Fuzzy Structural Systems Using Immune Network Modeling", Proceedings of the AIAA/ASME/ASCE/AHS SDM meeting, AIAA 99-1425, St. Louis, MO, April 1999, pp. 1880-1890.

20. Yoo, J., Soft Computing Adaptations in Optimal Design of Structural Systems, Ph.D. Thesis, Department of Mechanical Engineering, Rensselaer Polytechnic Institute, Troy, New York, 1999.

21. Haftka, R.T., and Gurdal, Z., 1993. Elements of Structural Optimization, Kluwer Academic Publishers, Dordrecht.

22. Michalewicz, Z., and Janikow, C.Z., 1991. "Handling Constraints in Genetic Algorithms", Proceedings of the 4th International Conference on Genetic Algorithms, pp. 151-157.

23. Hajela, P., and Lee, J., 1996. "Constrained Genetic Search Via Schema Adaptation: An Immune Network Solution", proceedings of the First World Congress of Structural and Multidisciplinary Optimization, Goslar, Germany, May-June, 1995, eds. N. Olhoff and GIN Rozvany, pp. 914-920, Pergamon Press.

24. Hajela, P. and Yoo, J., 1996. "Constraint Handling in Genetic Search Using Expression Strategies",AIAA Journal, Vol 34, No. 11, pp. 2414-2420.

25. Lin, C.-Y., and Hajela, P., 1993. "Genetic Search Strategies in Large Scale Optimization", proceedings of the 34th AIAA/ASME/ASCE/AHS/ASC SDM Conference, La Jolla, California, pp. 2437-2447.

26. Hajela, P., 1997. Stochastic Search in Discrete Structural Optimization - Simulated Annealing, Genetic Algorithms and Neural Networks, Discrete Structural Optimization, Springer, New York, pp. 55-134, (ed. W. Gutkowski).

27. Lee, J. and Hajela, P., "GA's in Decomposition Based Design - Subsystem Interactions Through Immune Network Simulation", Structural Optimization, vol. 14, No. 4, pp. 248-255, December 1997.

28. Hajela, P., "Biologically Inspired Computational Paradigms - Adaptive Solution Strategies for Multidisciplinary Structural Design", Invited Lecture at the 3nd International Conference on Computational Structures Technology, Budapest, Hungary, August-September, 1996, Advances in Computational Structures Technology, Civil-Comp Press, Edinburgh, Scotland, ed. BHV Topping, pp. 19-28.

29. Smith, R. E.; Forrest, S.; Perelson, A. S. 1992: Searching for Diverse Cooperative Populations with Genetic Algorithms. Technical Report CS92-3, University of New Mexico, Department of Computer Science, Albuquerque, NM.

30. Dasgupta, D., and Forrest, S., 1995. "Tool Breakage Detection in Milling Opeartions Using a Negative Selection Algorithm", TR No. CS95-5, Department of Computer Science, University of New Mexico.

31. Hajela, P., and Yoo, J., Immune Network Modeling in Design Optimization, New Methods in Optimisation, Editors: D. Corne, M. Dorigo and F. Glover, McGraw-Hill, 1999.

32. Richards, S.A., 1995. Zeroth-Order Shape Optimization Utilizing a Learning Classifier System, Ph.D. dissertation, Stanford University, California.

33. Hajela, P. and Kim, B., 1998. "Classifier Systems for Enhancing Neural Network Based Global Function Approximations", proceedings of the 7th AIAA/NASA/ISSMO/ USAF Multidisciplinary Analysis and Optimization Meeting, St. Louis Missouei.

34. Hajela, P., and Lee, J., "A Machine Learning Based

Approach to Improve Global Function Approximations for Design Optimization", proceedings of the International Symposium on Optimization and Innovative Design, JSME Paper No 116, edited by H. Yamakawa, M. Yoshimura, S. Morishita and M. Arakawa, July 28-July 30, 1997, Tokyo, Japan.

35. Goel, S., 1998. Machine Learning Paradigms in Design Optimization: Applications in Turbine Aerodynamic Design, Ph.D. dissertation, Rensselaer Polytechnic Institute, New York, April.

36. Wolfram, S., 1994. Cellular Automata and Complexity, Addison Wesley, New York.

37. Hajela, P., and Kim, B., " GA Based Learning in Cellular Automata Models for Structural Analysis", 3rd World Congress on Structural and Multidisciplinary Optimization, Niagara Falls, New York, May 1999.

38. Hajela, P., and Kim, B.,"On the Use of Energy Minimization for CA Based Analysis in Elasticity", submitted to the 41st AIAA/ASME/ASCE/AHS SDM Meeting, April 1-3, Atlanta, Georgia, 2000.

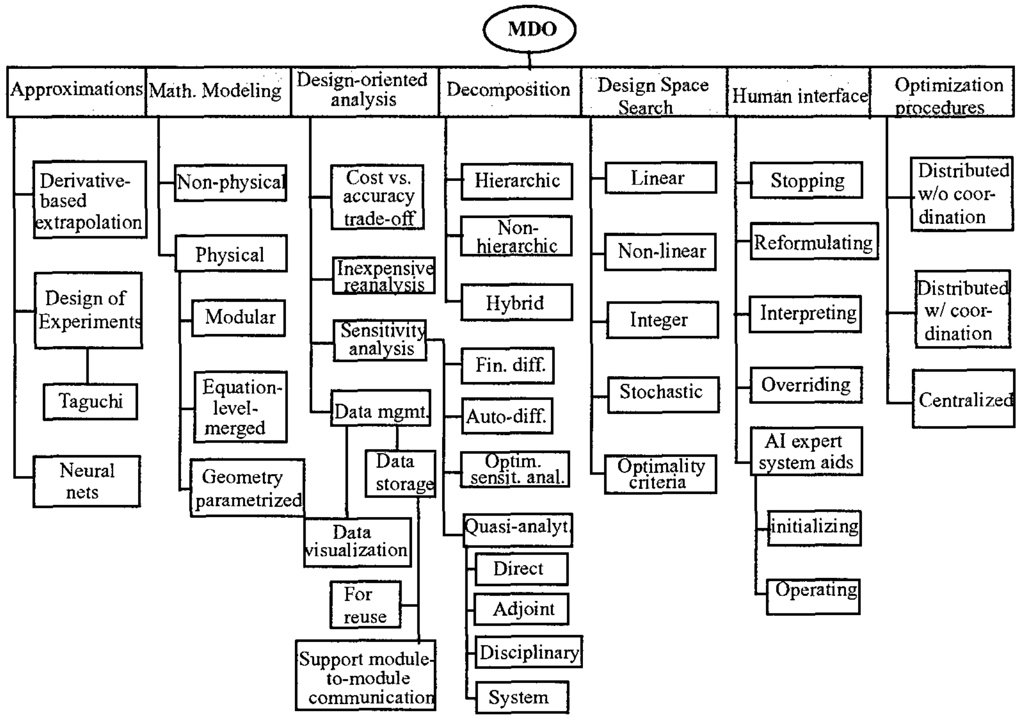


Figure 1. Taxonomy of the MDO Problem

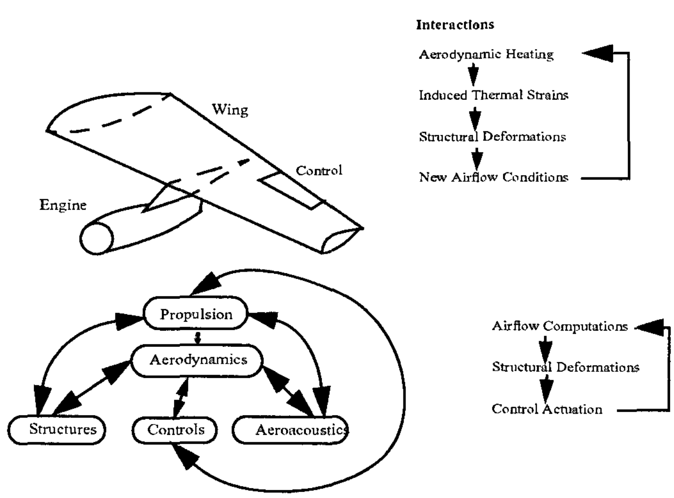


Figure 2. Schematic of a coupled MDO problem

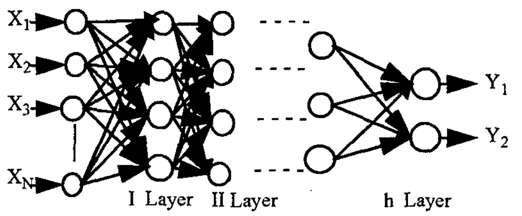


Figure 3. Schematic of a backpropagation neural network

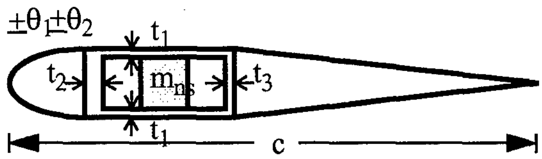


Figure 5b. Cross sectional profile of rotor blade

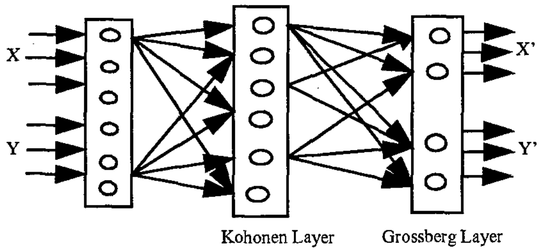
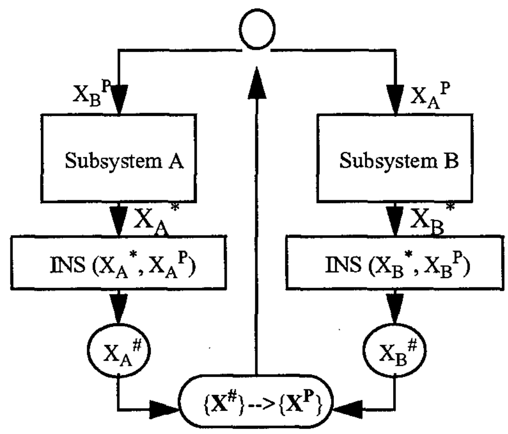


Figure 4. Schematic of a counterpropagation neural network



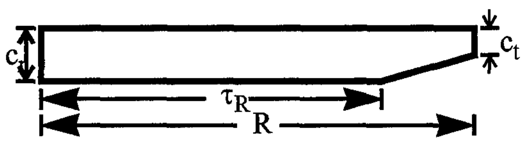


Figure 5a. Planform of rotor blade

Figure 6. Subsystem interactions through IN simulation

design variable		symbol
d_1	mass 1 location	m_1
d_2	mass 2 location	m_2
d_3	tuning mass 1 (kg)	d_{m1}
d_4	tuning mass 2 (kg)	d_{m2}
d_5	horizontal flange thickness ratio to box beam	t_1
d_6	left vertical flange thickness ratio to box beam	t_2
d_7	right vertical flange thickness ratio to box beam	t_3
d_8	blade twist (deg)	θ_t
d_9	twist shape parameter	δ
d_{10}	taper inception point	τ_R
d_{11}	chord ratio	λ_c
d_{12}	rotational speed (rad/sec)	Ω
d_{13}	layup angle of inner vertical flange (deg)	θ_1
d_{14}	layup angle of outer vertical flange (deg)	θ_2

Table 1. Descrition of the design variable set

shear force (N)			flap moment (N m)			lag moment (Nm)			thrust in cruise (N)			failure criterion		
NN	actual	%	NN	actual	%	NN	actual	%	NN	actual	%	NN	actual	%
13695	13546	1.10	11403	11711	2.08	5278	5266	0.24	778	784	0.76	0.261	0.261	0.01
8666	8649	0.19	6295	6891	0.50	4059	3994	1.65	517	506	2.17	0.243	0.241	0.83
6864	6790	1.09	6700	6745	0.67	4036	3880	4.72	369	374	1.33	0.247	0.249	0.84
9160	9373	2.27	8409	8406	0.04	3917	3772	3.84	498	502	0.79	0.313	0.308	1.62
14952	15028	0.51	12905	12984	0.61	4082	4086	0.10	857	873	1.83	0.250	0.250	0.02
17002	17001	0.03	12768	12736	0.25	5974	5899	1.27	926	921	0.54	0.319	0.316	0.94
7284	7339	0.75	6934	6957	0.33	11606	11559	0.40	435	427	1.87	0.238	0.244	2.45

Table 2. Testing of the trained BP network

shear force (N)			flap moment (N m)			lag moment (N m)			thrust in cruise (N)			failure criterion		
NN	actual	%	NN	actual	%	NN	actual	%	NN	actua l	%	NN	actual	%
8942	8687	2.93	7316	7504	2.53	3663	3705	1.13	472	483	2.28	0.25	0.258	0.39
14963	14272	4.84	10056	10264	2.02	8273	7892	4.82	766	763	3.93	0.312	0.312	0.01
12384	12579	1.55	10568	10458	1.05	5894	5758	2.36	763	754	1.19	0.235	0.237	0.84
8926	8919	0.08	7045	7058	0.18	5685	5713	0.49	509	502	1.39	0.264	0.266	0.75
10253	10156	0.96	13093	13185	0.69	11796	11183	5.48	933	907	2.86	0.329	0.333	1.20
8113	8046	0.84	8936	8841	1.01	12529	11932	5.00	376	371	1.35	0.347	0.349	0.57
9735	9653	0.85	9618	9685	0.69	22940	23695	3.18	467	454	2.86	0.394	0.390	1.02

Table 3. Testing of the trained CP network

	subsystem [A]	subsystem [B]	subsystem [C]
objective	$F(\boldsymbol{X}) = c_1 F_z + c_2 M_y + c_3 M_z$		
constraints	$HP_h \leq HP_a$ $\eta^L \leq \eta \leq \eta^U$ $AI \geq AI^L$	$HP_f \leq HP_a$ $C_T/\sigma^L \leq C_T/\sigma \leq C_T/\sigma^U$	$W_b \leq W_b^U$ $\bar{R} \leq 1$ $\sigma_{buck} \leq \sigma_{all}$
design variables	$m_1, m_2, m_3, t_1^4, t_1^5, t_1^6, t_2^3, t_2^6, t_2^9, t_3^6, t_3^8, t_3^9, \tau, \pm\theta_1$	$t_1^3, t_1^{10}, t_2^4, t_2^5, t_2^7, t_2^8, t_3^2, t_3^3, t_3^5, t_3^7, t_3^{10}, \theta_t, \delta, \pm\theta_2$	$m_4, m_5, t_1^1, t_1^2, t_1^7, t_1^8, t_1^9, t_2^1, t_2^2, t_2^{10}, t_3^1, t_3^4, \lambda_c, \Omega$

Table 4 Topology of system decomposition

DISCUSSION

Session III, Paper #17

Mr Templin (NRC, Canada) observed that advanced search tools are potentially very complex and therefore perhaps require "specialist" users. He wondered whether the author saw the need for such people, or whether he anticipated another solution.

> **Prof Hajela** suggested that as the toolsets are changing, so is the education process. He believes that today's disciplinary-oriented graduates probably already have some training in advanced search methods. He further believes that it is these graduates who will lead industry in a "forced migration" towards at least some of these approaches.